A Performance History of Several Multitasking Codes on the NAS Y-MP

Robert J. Bergeron[1]

Report RND-91-008, July 1991

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

A Performance History of Several Multitasking Codes on the NAS Y-MP

Robert J. Bergeron[1]

Report RND-91-008, July 1991

RND Branch
NAS Systems Division
NASA Ames Research Center
Mail Stop 258-6
Moffett Field, CA 94035-1000

---

[1] Computer Sciences Corporation, NASA Contract NAS 2-12961, Moffett Field, CA 94035

A PERFORMANCE HISTORY OF SEVERAL MULTITASKING CODES ON THE NAS Y-MP

Robert J. Bergeron
Computer Science Corporation
NASA Ames Research Center
Numerical Aerodynamic Simulation Systems Division

ABSTRACT

This paper reports performance histories of codes used to ensure that the Y-MP maintains a sustained multitasking capability across the major Cray upgrades. The history of the ARC3D code shows that this capability should be tested to ensure that upgrades do not impair performance. Measurements indicate that Cray hardware and software upgrades have improved the performance of some multitasking codes as much as 13% in the $2\frac{1}{2}$-year history described herein.

INTRODUCTION

This paper reports performance histories of codes used to ensure that the Cray Y-MP maintains a sustained multitasking capability across the major Cray upgrades. Two of the codes, ARC3D and THRED, provided verification that the Y-MP could sustain 1 GFLOP/sec during the NAS acceptance testing of Y-MP Serial Number (SN) 1002 (Barton, 1989). Recent upgrades have motivated the addition of two other codes to this history. The third code, SPARK, provides useful data on the state of the autotasker and a fourth code, MXM, provides information on the efficiency of the operating system with respect to autotasking.

The above codes measure the ability of Cray upgrades to maintain efficient parallel performance at the DO loop level. Cray supercomputers require compiler directives to distribute iterations of Fortran DO loops among the available processors. Cray terms insertion of these directives by the user as microtasking and insertion of the directives by a preprocessor as autotasking (Cray, 1989). Cray, beginning with the 6.0 level of UNICOS, provided two paths for DO loop level parallel performance. The first path, from UNICOS 5.1, employs hardware semaphore synchronization to obtain the maximum parallel performance. The hardware semaphore keeps all slave processes attached to the master even during periods of singletasked work. The second, or 6.0, path provides software

to assist in process synchronization for optimal parallel workload performance and this path is the one travelled by the user under normal operating circumstances. The software semaphore returns to the operating system any CPUs idled by singletasked work. The 6.0 software path reduces the number of interrupts and system overhead, but also reduces the parallel performance since the master task must more frequently request CPUs from the operating system to perform parallel work. This report will record the performance of the SPARK code under both the 5.1 and 6.0 paths.

In addition to software upgrades involving the compiler, autotasker, and operating system, Cray provided a hardware upgrade in the form of a new machine, SN 1030. This machine displayed a reduced clock cycle and expanded common memory. The new clock cycle was 5.5% less than that of SN 1002.

This monitoring effort began on a systematic basis only with the UNICOS 6.0 upgrade; therefore, some of the chronologies presented below may have gaps. The monitoring emphasizes performance on major upgrades, that is, new versions of the operating system and compiler. All results reported herein reflect calculations performed on a dedicated machine. The data below reflect the current monitoring; revisions to this report will reflect performance changes due to future Y-MP upgrades.

ARC3D

ARC3D solves the three-dimensional unsteady thin layer Navier-Stokes equations in generalized curvilinear coordinates (Pulliam and Steger, 1989). The solution method is an implicit approximate factorization algorithm with explicit boundary equations. The code consists of time steps involving separate sweeps in each of three directions; each 1-D sweep includes a setup and a block solver for each plane. The block tridiagonal solver employed in this code requires over 50% of the CPU time.

The original version of ARC3D required 56 MW of common memory. To conduct the Government-required Acceptance Test on the 32 MW SN 1002, Cray Research transferred some of the common memory array storage to disk. In addition, the Government required a 1 GFLOP/sec performance on the modified code.

Therefore, Cray recast the original code into a 26 MW version which outperformed the original code by 20% (e.g., 184 MFLOP/sec vs 144 MFLOP/sec) in singletasked mode. Cray changes involved loop optimization in major subroutines and storage modifications allowing faster access to array data. Since about 30 MW of data resided on disk, ARC3D required a high speed data transfer in a small elapsed time to produce high performance in dedicated time.

To produce a parallel version, Cray decomposed each plane into slabs and employed microtasking to perform each setup and solver in parallel. This is the version of ARC3D used for the following history.

The Y-MP Acceptance Test discussions established the total GFLOPS for a 200 timestep ARC3D execution as 1199 (Subsequent measurements with the Y-MP Hardware Performance Monitor have established a more accurate count of 1212 GFLOPS). The 8 CPU

Y-MP executed the program in 1114 seconds for a performance rate of 1.076 GFLOP/sec. For the Acceptance Test, the SSD served as a file system to facilitate high speed data transfer between it and central memory. The Appendix discusses the various SSD arrangements for ARC3D.

The table below contains a chronology of the ARC3D testing and Figure 1 provides a plot of the performance data. Some gaps exist and some of the minor upgrades (changing second or third digit) may actually employ kernel mods intended for the major upgrade.

Date	FPP	FMP	Compiler	UNICOS	Seconds	GFLOP/sec	Comments
10/13/88	N/A	N/R	3.0	4.0.6+	1114	1.076	1002 Acceptance Test
06/29/89	N/A	N/R	3.0	5.0.8	1525	0.786	UNICOS problems
11/02/89	N/A	N/R	3.0.1.16	5.0.12	1543	0.777	UNICOS problems
11/08/89	N/A	N/R	3.0.1.16	5.0.12	1100	1.090	Cray test SDS
11/08/89	N/A	N/R	3.0.1.16	5.0.12	1164	1.030	Cray test SSD
02/03/90	N/A	N/R	3.0.2.2	5.0.13	1086	1.104	1030,5.5% faster clock
04/17/90	------	---------	--------	5.1.8	----	-----	5.1 upgrade
11/01/90	------	---------	--------	6.0.1	----	-----	6.0 upgrade
03/13/91	N/A	4.0.3(9)	4.0.3.1	6.0.9	1147	1.045	6.0 File Structure
04/24/91	N/A	5.0 (29)	4.0.3.1	6.0.11	1036	1.157	NAS test with SDS

N/A-Not Applicable
N/R-Not Recorded

In June of 1989, an attempt to duplicate the 1 GFLOP/sec performance of ARC3D with a kernel containing UNICOS 5.1 updates produced a performance rate of 0.786 GFLOP/sec. ARC3D represents a class of NAS Y-MP codes which require high speed I/O to achieve high floating point performance. A parallel version of the code consists of master and slave tasks. The slave tasks may be disconnected during periods for which the master is performing I/O. These tasks require reconnection when the calculation resumes or the calculation will proceed in singletask mode. The performance degradation occurred because a modification contained in the 5.1 update did not check frequently enough (every minor clock tick) for reconnection of microtasked processes. This error is the source of the performance decrease in Figure 1 and would afflict all ARC3D-class codes executing in parallel mode in a dedicated environment.

Cray claimed to have corrected this problem about a month after notification by NAS. The next NAS test, run in November of 1989, failed to give a 1 GFLOP/sec performance when run with the SSD in its NAS default configuration as ldcache. A subsequent 1.090 GFLOP/sec execution with SSD configured as Secondary Data Segments (SDS) by Cray indicated that the NAS default configuration and not the 5.1 modification was the problem. Cray then increased the NAS default block size to obtain a 1.030 GFLOP/sec rate with the SSD configured as ldcache.

The Y-MP Acceptance Test for SN 1030 saw ARC3D perform at 1.104 GFLOP/sec with the SSD configured as a file system.

In March of 1991, ARC3D performed the calculation in 1147 seconds with the I/O performed as before, i.e., with unformatted pure files on the SSD. The file structure of UNICOS 6.0 differs from the 5.1 structure and this difference has been suggested as the reason for the 6% degradation. Since a subsequent test with the SSD configured as SDS produced a performance rate of 1.157 GFLOP/sec, the current 6.0 operating system supports multitasking quite adequately. System upgrades normally produce performance fluctations, but only a performance rate below 1 GFLOP/sec should cause concern for NAS. Future testing should employ the SDS configuration to avoid problems with the file structure formats.

The compilation provided in the table indicates an 8% improvement in ARC3D Y-MP performance since the NAS Acceptance Test of SN 1002. Since the most recent execution of ARC3D required 7099 CPU seconds on a single processor, the parallel version of ARC3D currently displays a speedup of about 6.8.

THRED

THRED performs a full three-dimensional temporally evolving simulation of mixing of reactive fluids (Claus, 1988). Pseudospectral methods are used to compute all derivatives. Although the fluid flow is assumed incompressible, terms are added for a variable density due to the mildly exothermic nature of the chemical reactions.

For the SN 1002 Acceptance Test, Cray recast the original 25 MW code which performed at 148 MFLOP/sec on one CPU into a parallel form which performed at 169 MFLOP/sec on one CPU. Cray vector changes involved temporary variables in major loops to reduce memory access.

The parallel version employs microtasking to perform much of the FFT calculation in parallel. The microtasked version of THRED contains a parameter NCPU which specifies the number of CPUs for the calculation. A second Cray parameter NSEG distributes (stripmines) the DO loop computational work equally to the CPUs requested. Execution with a different number of CPUs requires recompilation. This is the version of THRED used for the following history.

Discussions during the Acceptance Test established the total GFLOPS for a 400 timestep version of THRED as 1494. (A subsequent execution of THRED with the HPM indicated the total number of floating point operations as 1333 or 10% less than the Acceptance Test value). The Cray version of THRED performed this calculation on 8 CPUs in 1218 seconds for a performance rate of 1.227 GFLOP/sec. At the system upgrade Acceptance Test, THRED performed the calculation in 1080 seconds. On March 13 1991, THRED performed the calculation in 1057 seconds.

The table provides a chronology of the THRED testing.

Date	FPP	FMP	Compiler	UNICOS	Seconds	GFLOP/sec	Comments
10/13/88	N/A	N/R	3.0	4.0.6+	1218	1.227	1002 Acceptance
06/29/89	N/A	N/R	3.0	5.0.8	1190	1.255	OK-no clock pblm
02/03/90	N/A	N/R	3.0.2.2	5.0.13	1080	1.383	1030, 5.5% faster clock
04/17/90	------	---------	--------	5.1.8	----	-----	5.1 upgrade
11/01/90	------	---------	--------	6.0.1	----	-----	6.0 upgrade
03/13/91	N/A	4.0.3(9)	4.0.2.1	6.0.9	1057	1.413	6.0 File Structure

N/A-Not Applicable
N/R-Not Recorded

The Y-MP was able to sustain 1 GFLOP/sec on THRED in spite of its inability to deliver 1 GFLOP/sec on ARC3D. Unlike ARC3D, the THRED code does not require continuous high-speed I/O and subsequent timely reconnection of slave processes to achieve 1 GFLOP/sec.

The table indicates a 13% improvement in THRED Y-MP performance since the NAS Acceptance Test of SN 1002. Since the most recent execution of THRED required 7862 CPU seconds on a single processor, the parallel version of THRED currently displays a speedup of about 7.4.

SPARK

SPARK models the behavior of a two-dimensional supersonic chemical reacting mixing layer (Drummond, et. al., 1987). The 8 MW code employs a modified MacCormack (partially implicit predictor-corrector scheme) technique to solve the Navier-Stokes equations. A multicomponent finite-rate chemistry model describes the chemical reactions.

Subroutines composed of single Fortran DO loops containing hundreds of floating point operations per iteration distinguish SPARK from other NAS codes; in addition, the subroutine performing the calculation of reaction rates consists of several unrolled loops. The code performs at about 185 MFLOP/sec in singletasked mode. However, the autotasker requires directives to autotask the single DO loop subroutines. Cray's model assumes a nested DO loop and the autotasker generally tries to stripmine this construction by sending the inner vectorized loop as a group to each CPU. The SPARK version with the autotasking directives inserted was used for the following history.

The table below contains a chronology of the SPARK testing and Figure 2 provides a plot of the performance data. This testing began about the same time Cray introduced autotasking. Problem time has been recently increased to ensure that degradations in parallel performance produce noticeable results (i.e., to prevent the tendency to dismiss 1 or 2 second deviations as insignificant). The large variation shown in Figure 2 for the performance of SPARK in 1990 reflects the change in the autotasker. In contrast, Figure 1 shows a relatively constant performance for the microtasked ARC3D during this period.

Date	FPP	FMP	Compiler	UNICOS	Seconds	GFLOP/sec	Comments
5.1 Option							
12/14/89	2.26B18	N/R	3.0.2.2	5.0.13	18.18	0.700	12.73 GFLOPS
12/14/89	2.24S6	N/R	3.0.2.2	5.0.13	20.39	0.624	
12/14/89	2.24S6	N/R	3.0.2.2	5.0.13	15.37	0.828	Old Libraries
07/13/90	2.24S6	N/R	3.0.2.2	5.1.10	14.49	0.879	
07/13/90	3.00Z51	N/R	4.0.	5.1.10	18.48	0.689	
11/01/90	-------	-------	--------	6.0.1	-----	-----	6.0 upgrade
11/06/90	3.00Z36	4.0.0	4.0.1	6.0.11	16.01	0.795	
04/25/91	3.00Z61	4.0.3	4.0.3.1	6.0.11	151.1	0.732	110.67 GFLOPS
04/25/91	3.03Y4	5.0 (29)	5X402417	6.0.11	142.2	0.778	
6.0 Option							
04/25/91	3.00Z61	4.0.3	4.0.3.1	6.0.11	150.8	0.734	
04/25/91	3.03Y4	5.0 (29)	5X402417	6.0.11	143.7	0.770	

N/R-Not Recorded

The table indicates a 12% increase in SPARK performance since the introduction of autotasking. Software accounts for about 7% of this improvement. SPARK currently displays a speedup of 4.2 under the 5.1 option and a speedup of 4.1 under the 6.0 option. The table also indicates a peak in SPARK performance on last release of FMP version 2. Examination of the listing indicates FPP version 2 parallelized more DO loops than version 3 and these transformations produced safe results for SPARK. Version 3 of the autotasker was less aggressive in loop transformations, apparently in response to user complaints.

MXM

MXM is a 50 MW test program which performs an unrolled matrix multiply. This version, taken from NASKERN (Bailey and Barton, 1985), measures the ability of the autotasker and operating system to deliver efficient parallel performance as the number of processors involved in the calculation increases.

MXM calls the system wall clock routine to report the elapsed time for the matrix multiplication only. The autotasker should not be allowed to substitute library routines for the matrix multiply. For an n by n matrix, the algorithm requires $2n^3$ FLOPS. The algorithm requires $2.5n^3$ memory references, but hardware monitor measurements indicate the actual memory references as $1.5n^3$.

The table provides a chronology of MXM testing. Matrix size has increased from the original 512 to 4096 to allow use with faster supercomputers.

Date	FPP	FMP	Compiler	UNICOS	Seconds	GFLOP/sec	Comments
01/30/91	2.26B18	N/R	3.0.2.2	6.0.9	0.12	2.278	n=512
01/30/91	2.24S6	N/R	3.0.2.2	6.0.9	59.79	2.302	n=4096
02/13/91	3.00Z61	4.0.3 (9)	4.0.3.1	6.0.11	59.79	2.302	n=4096
04/28/91	3.03Y4	5.0 (29)	5X402417	6.0.11	59.91	2.297	n=4096

N/R-Not Recorded

The table below shows the measured efficiency as a function of the number of CPUs requested for processing. A linear fit to the data allows the extrapolation of efficiency to 16 processors; the extrapolation may be optimistic.

Efficiency vs NCPUS

Date	NCPUS								
	1	2	3	4	5	6	7	8	Comments
01/30/91	1.000	0.999	0.992	0.991	0.991	0.988	0.983	0.989	E=0.969 for 16 CPUs
02/13/91	1.000	0.994	0.994	0.997	0.996	0.992	0.991	0.987	E=0.978 for 16 CPUs

Figure 3 plots this data up to and including 8 CPUs. Since the data reflect the matrix multiply times only, departures from 1.0 reflect the overhead of process synchronization and load balancing effects.

Synchronization overhead should increase linearly with the number of CPUs and such an increase should produce a linear decrease in the efficiency. The fraction of time spent in the serial part of the code is about 0.002 and increases very slowly for the larger matrix.

Figure 3 shows the load imbalancing as dips in the efficiency curve for the 512-element matrix (01/30/01), especially at 3 and 7 NCPUs. Load imbalancing effects appear smaller for the larger 4096-element matrix.

The figure shows the extrapolation to 16 CPUs as dashed lines without data points. The extrapolation indicates autotasking efficiency should degrade slightly as the number of processors is increased to 16.

The current 8-processor speedup for MXM is 7.9

CONCLUSION

This paper reports performance histories of codes used to ensure that the Y-MP maintains a sustained multitasking capability across the major Cray upgrades. The history of the ARC3D code shows that this capability should be tested to ensure that upgrades do not impair performance. Measurements indicate that Cray upgrades have improved the performance of some multitasking codes as much as 13% in the $2\frac{1}{2}$-year history described herein. Software improvements have produced a little over half of the performance increase.

References

Bailey, D. H., and Barton, J. T., 1985. The NAS Kernel Benchmark Program, NASA Technical Memorandum 86711, August 1895.

Barton, J. T., 1989. Early experience with the NAS Y-MP. In Proc. 22nd Semi-Annual Cray User Group Mtg, pp 116-122.

Claus, R. W., 1988. Time-accurate Simulations of a Shear Layer Forced at a Single Frequency, Paper 88-0061,26th AIAA Mtg, Reno, Nv., Jan 11-14,1988.

Cray Research Inc, 1989. UNICOS Autotasking User's Guide.

Drummond, J. P., Rodgers, R. C., and M. Yousuff Hussaini, 1987. A Numerical Method for Supersonic Reacting Mixing Layers. In Computer Methods in Applied Mechanics and Engineering, 64 1987.

Pulliam T. H., and Steger, J.P., 1978. Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow. In AIAA J. 18, pp. 159-167 (1978).

Figure 1

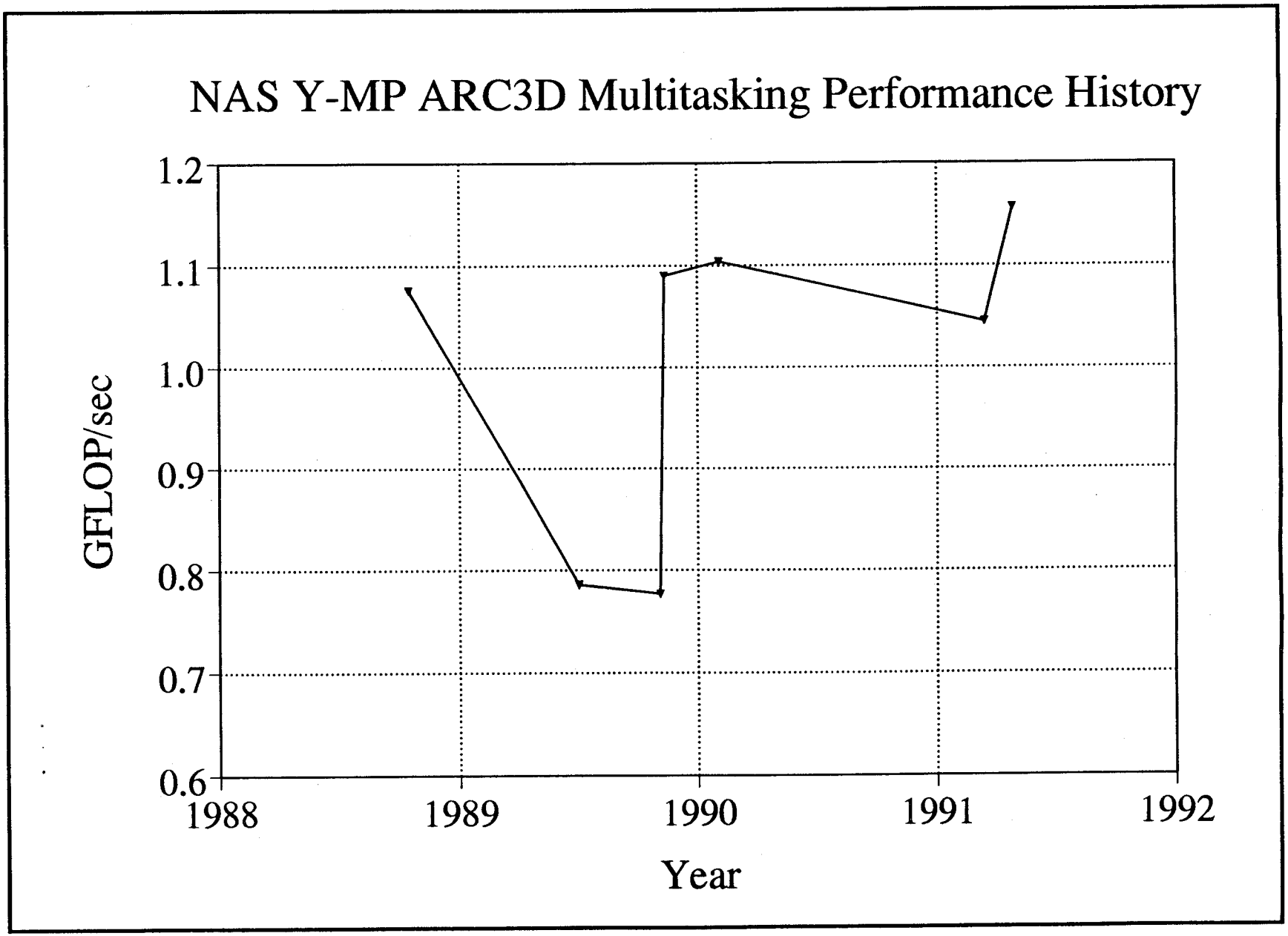

NAS Y-MP ARC3D Multitasking Performance History
GFLOP/sec
1.2
1.1
1.0
0.9
0.8
0.7
0.6
1988
1989
1990
1991
1992
Year

Figure 2

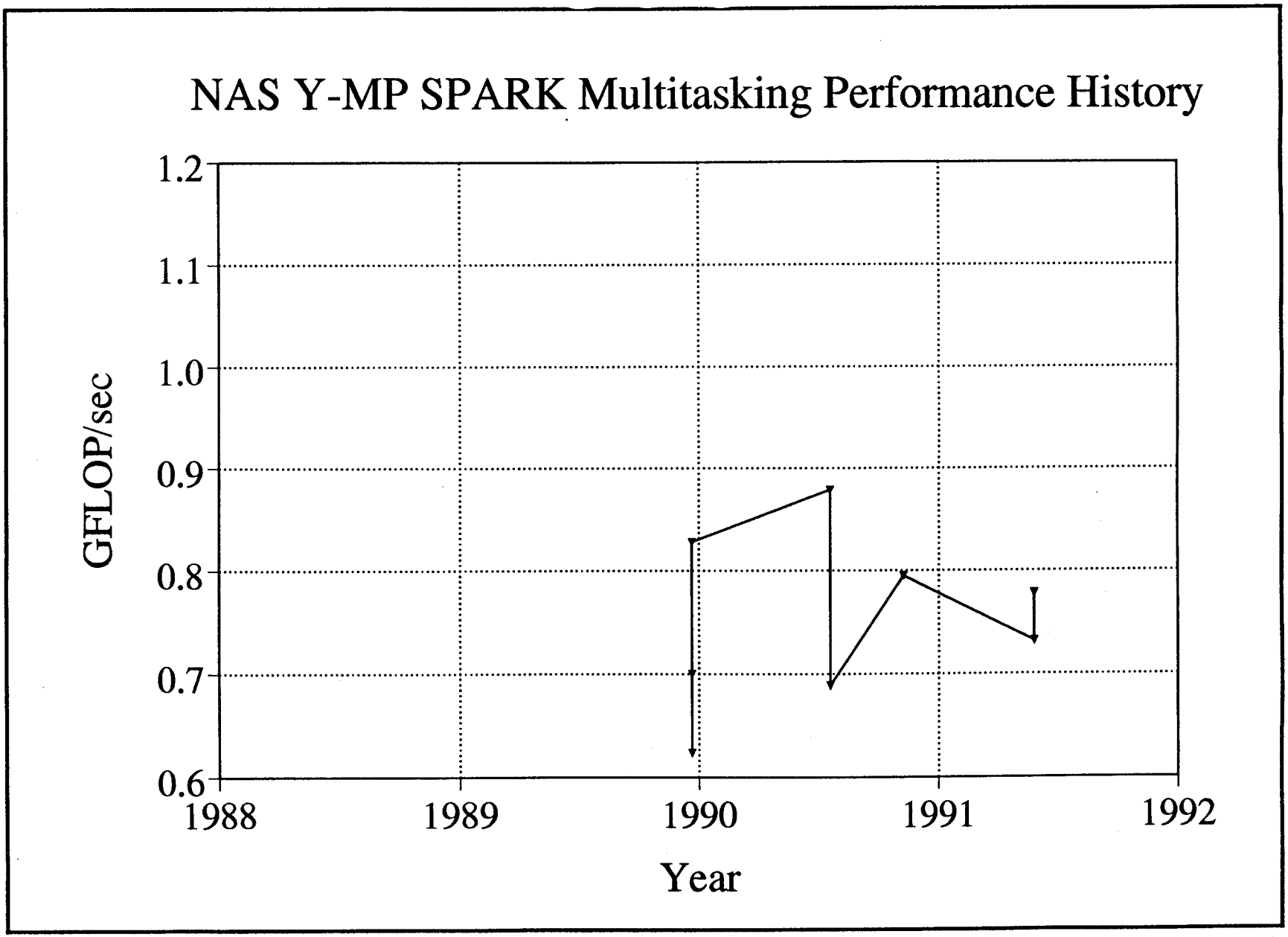
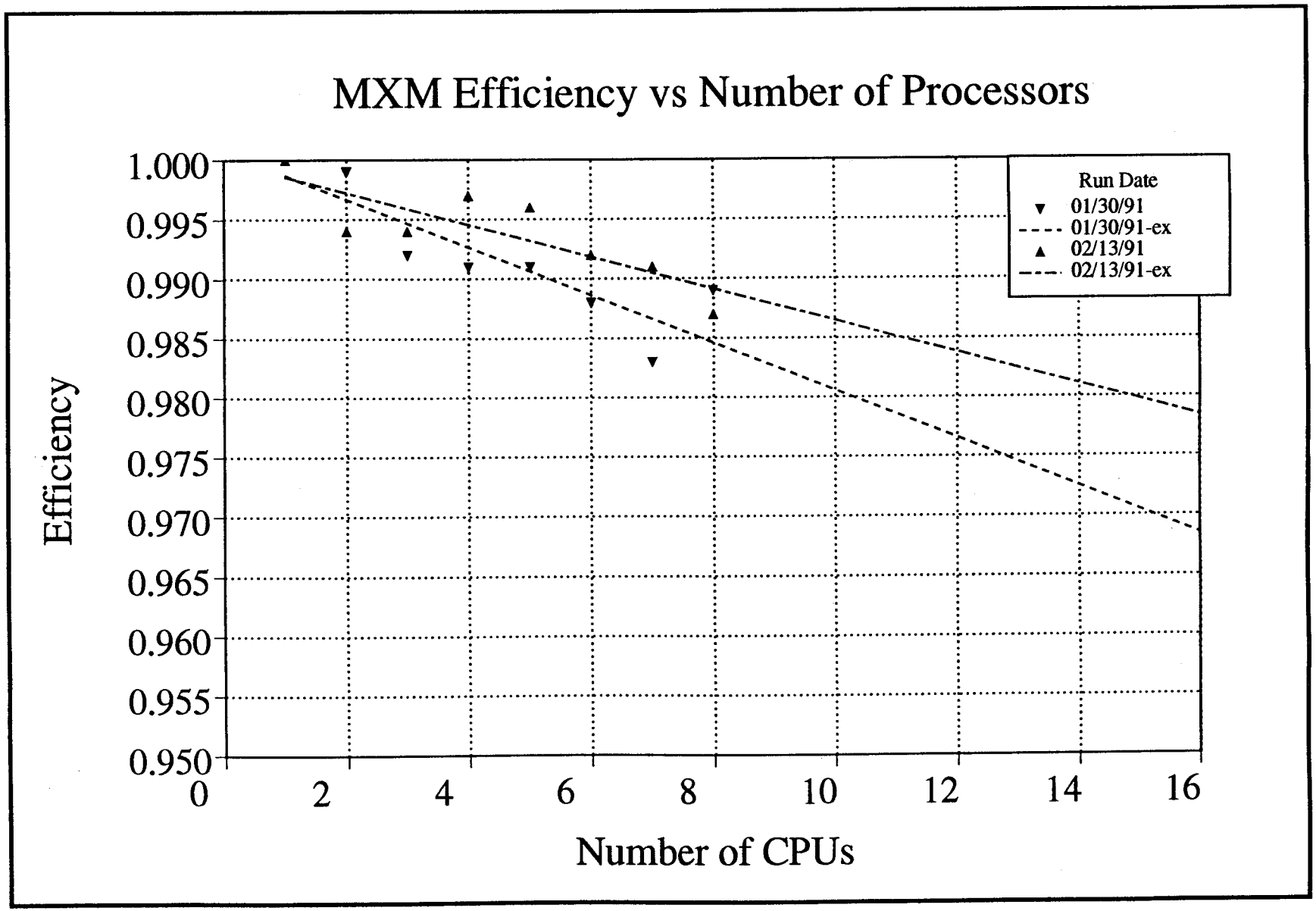

NAS Y-MP SPARK Multitasking Performance History
GFLOP/sec
1.2
1.1
1.0
0.9
0.8
0.7
0.6
1988
1989
1990
1991
1992
Year

Figure 3


MXM Efficiency vs Number of Processors
Efficiency
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0.955
0.950
0
2
4
6
8
10
12
14
16
Number of CPUs
Run Date
01/30/91
01/30/91-ex
02/13/91
02/13/91-ex